

FIG. 1A

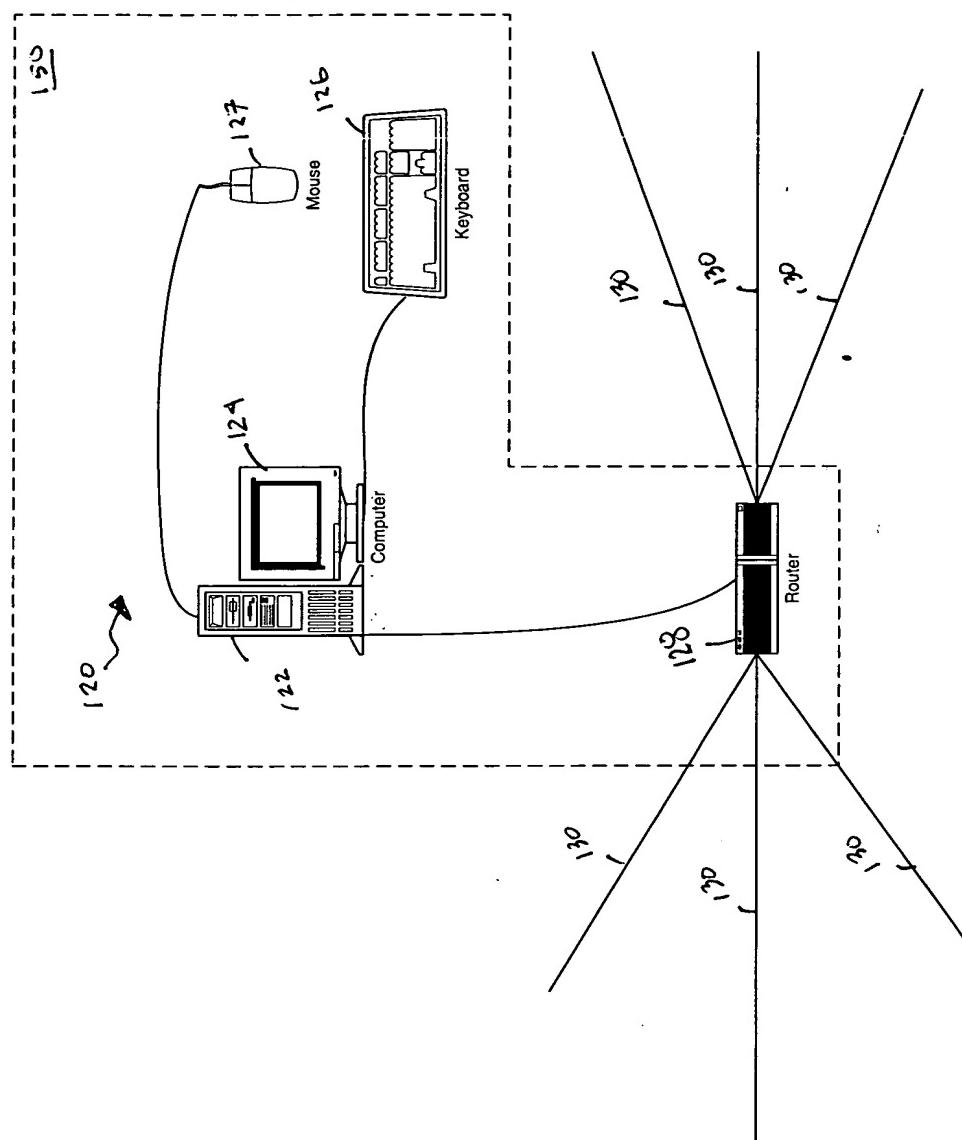


FIG. 1B

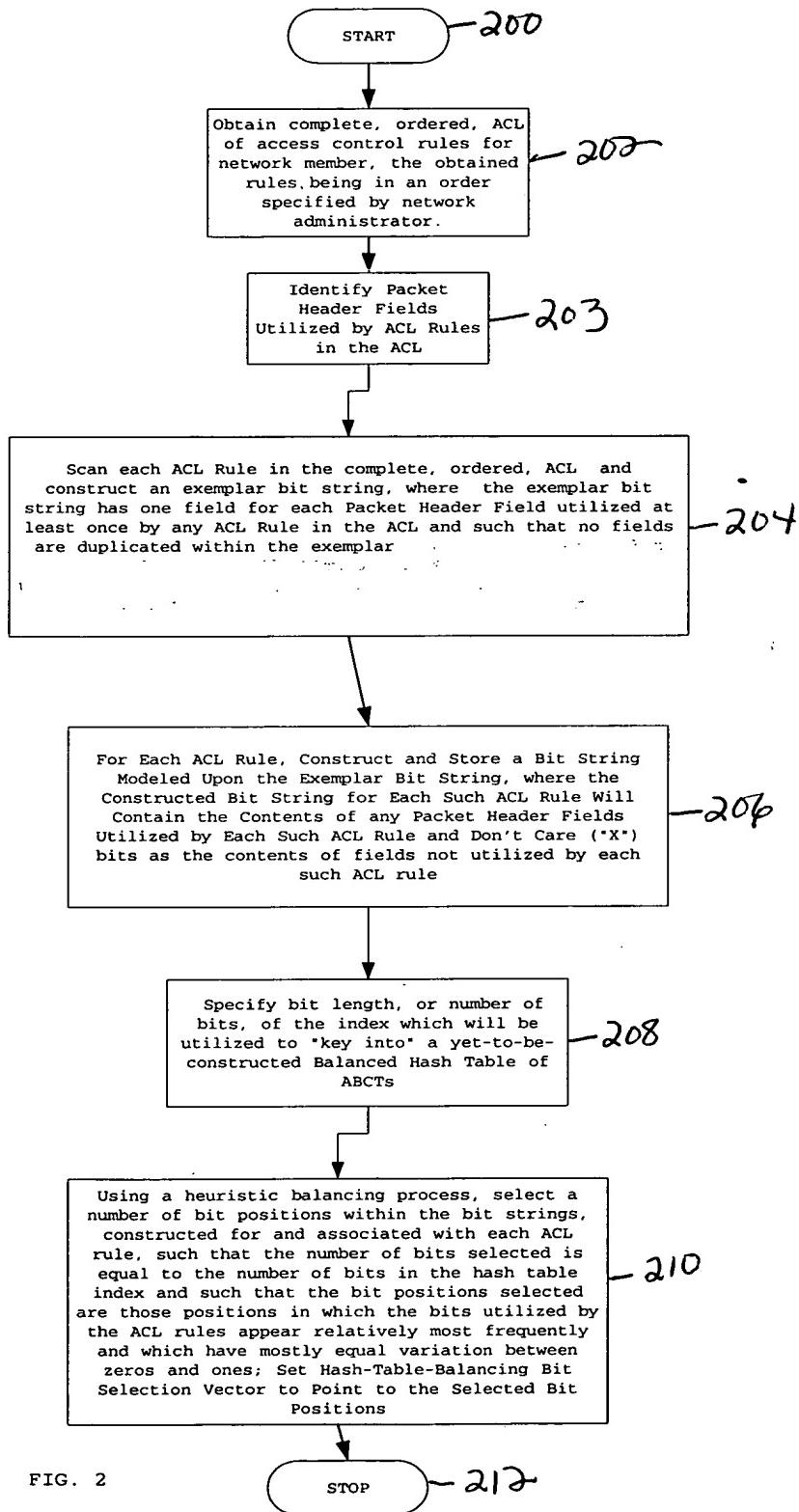


FIG. 2

STOP - 212

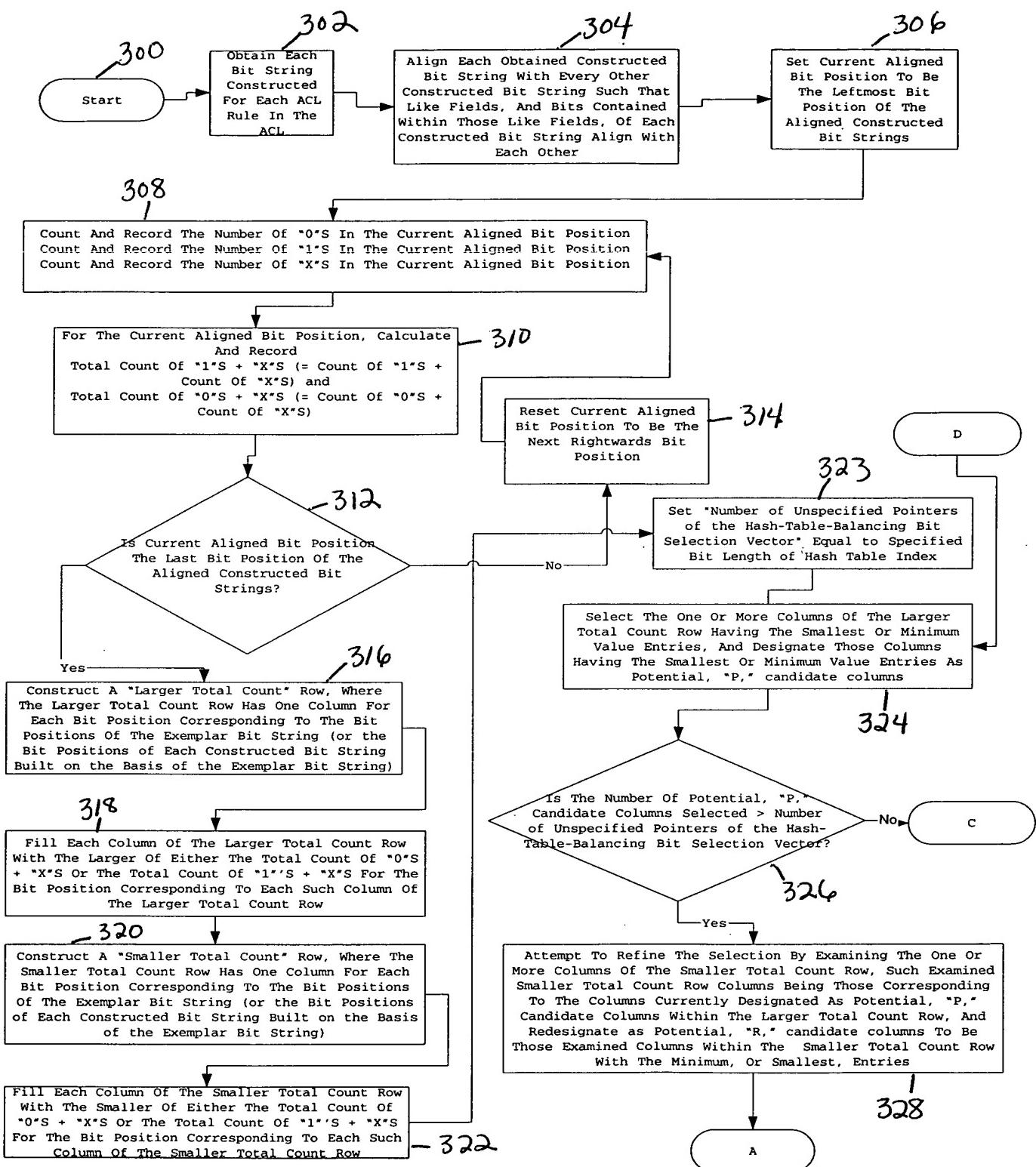
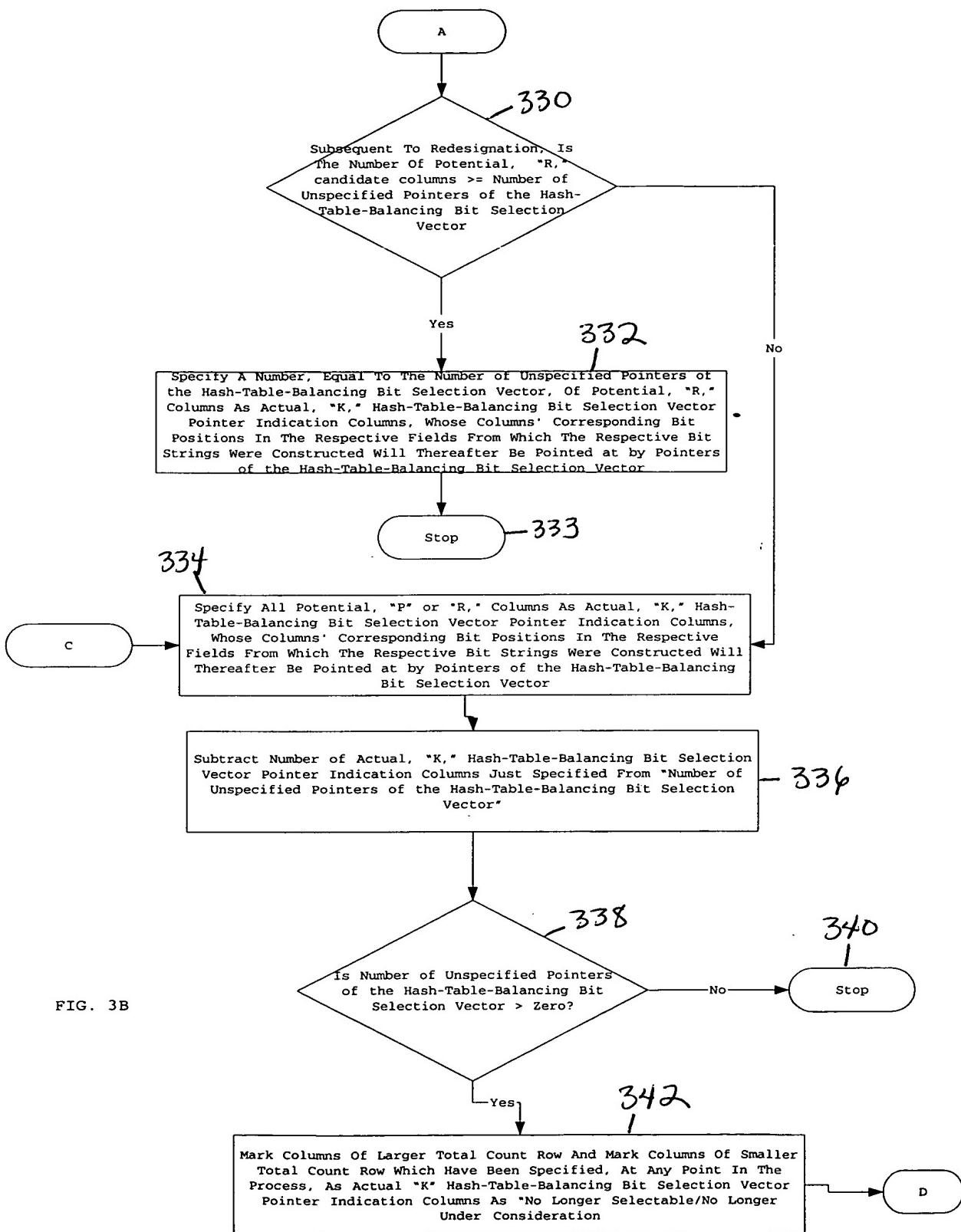
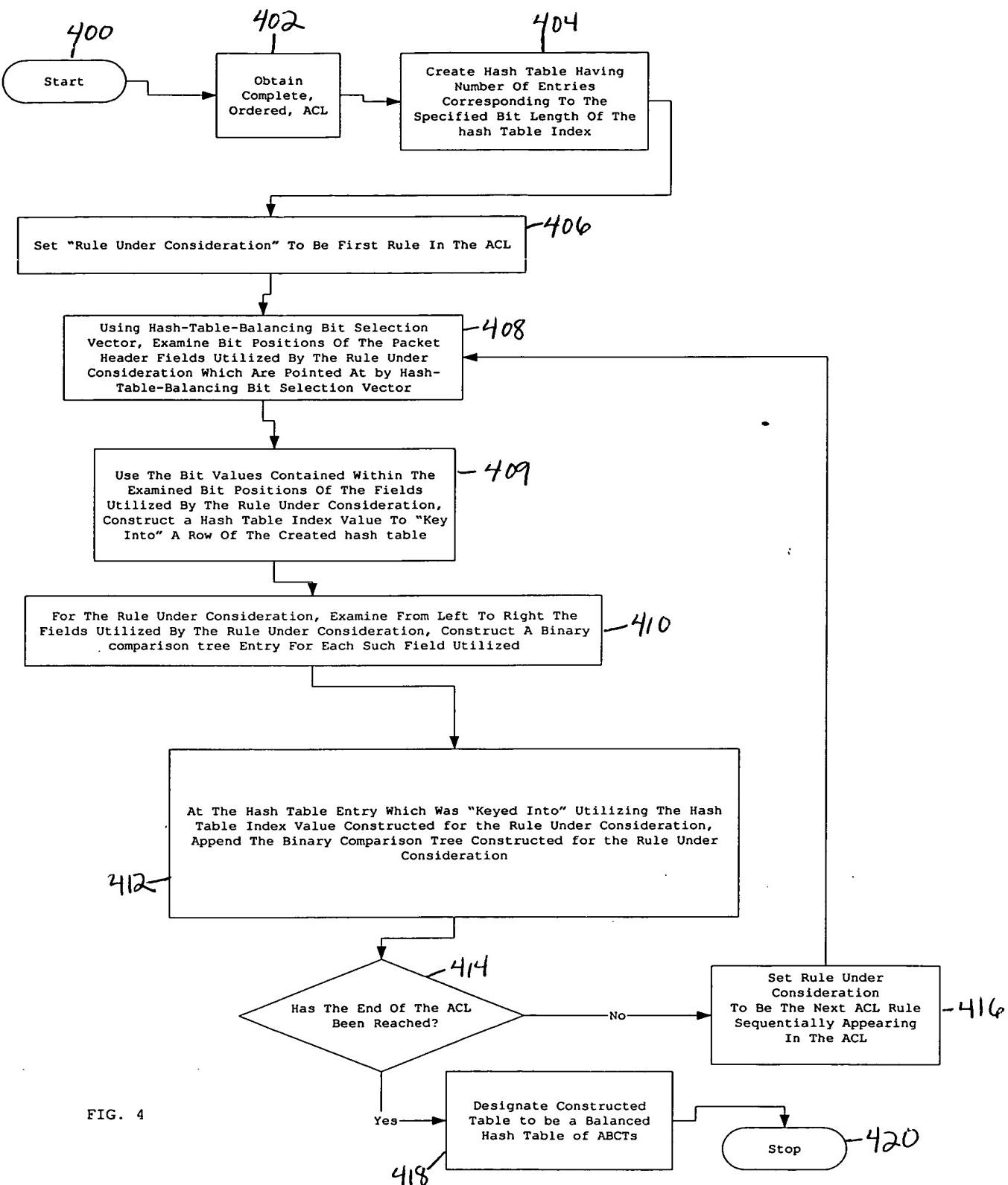
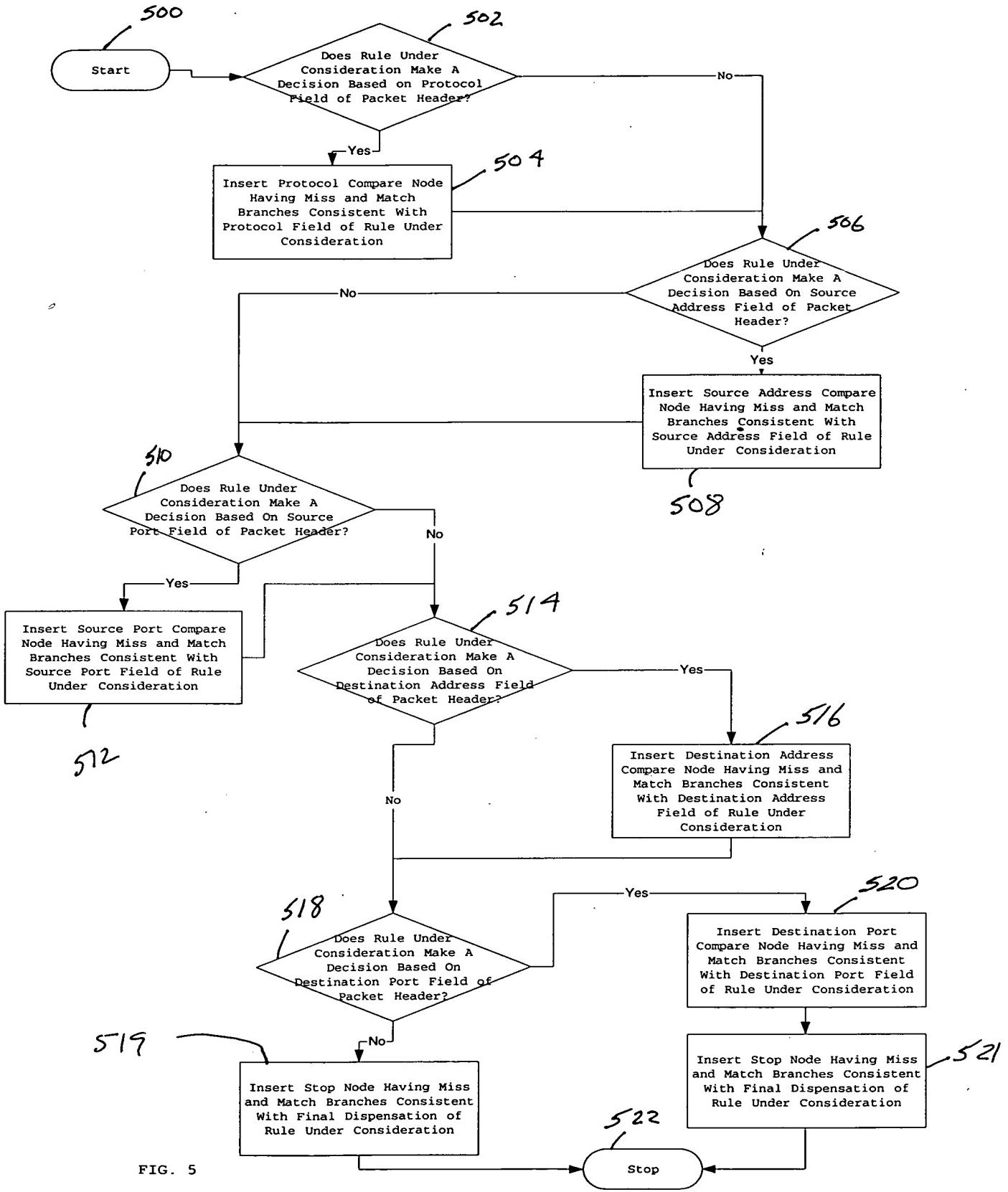


FIG. 3A







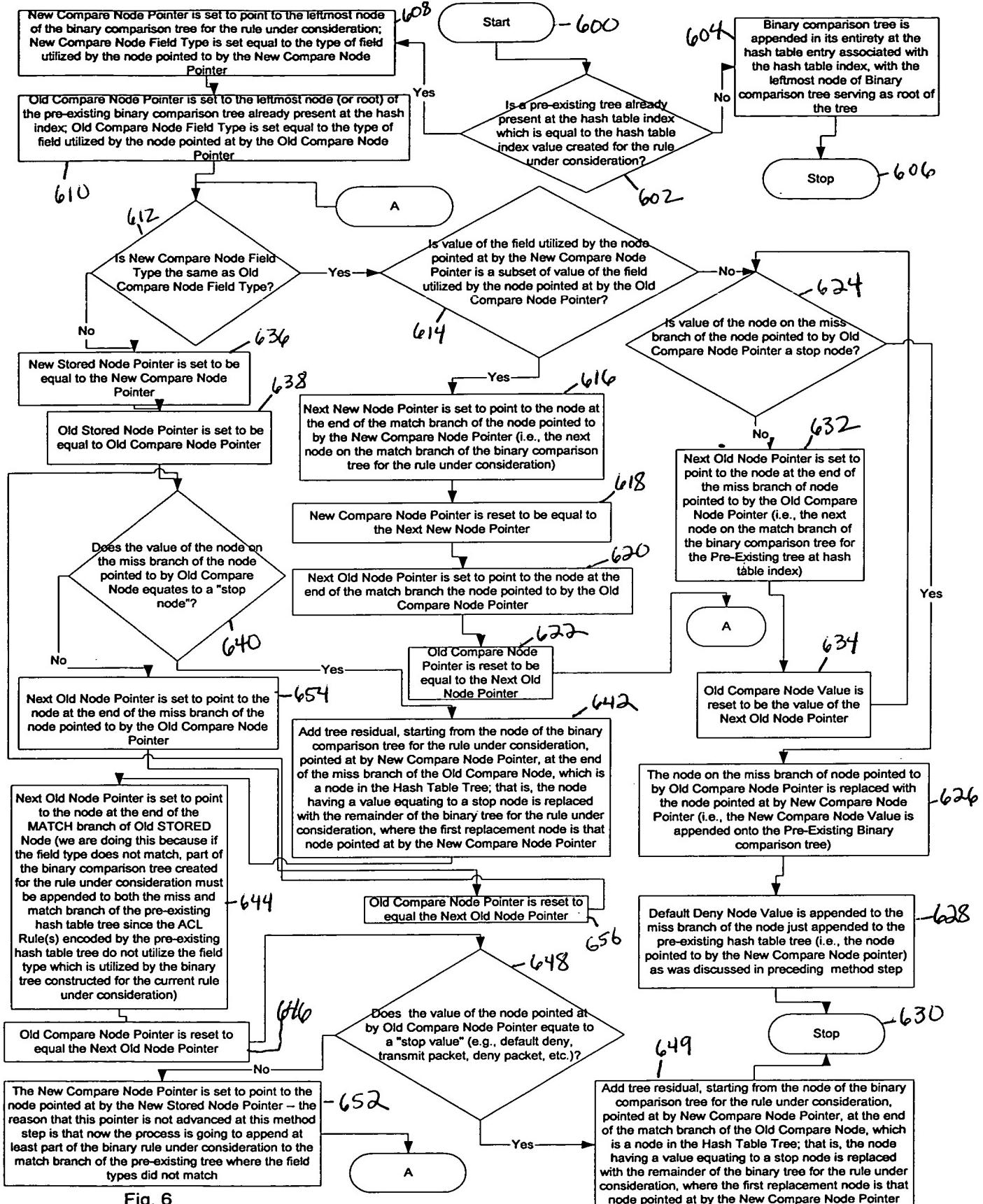


Fig. 6

Simplified Example of Ordered ACL Rule Set Typically Entered by a Network Administrator	
ACL Rules in an Ordered ACL Rule Set Expressed as Plain English Statements	Examples of Coded Versions of ACL Rules Which Are Typically Utilized Within an ACL Rule Set
Permit TCP protocol packets from any source IP address going to host having an IP address of 28.16.31.10 and a port identifier equal to 28.	PERMIT TCP ANY HOST 28.16.31.10 EQ 28
Deny TCP protocol packets from any source IP address going to host having an IP address of 28.16.31.10 and a port identifier greater than 23.	DENY TCP ANY HOST 28.16.31.10 GT 23
Deny UDP protocol packets from any source IP address going to host having an IP address of 30.22.12.5 and a port identifier equal to 11.	DENY UDP ANY HOST 30.22.21.5 EQ 11
Permit UDP protocol packets from any source IP address going to host having an IP address of 30.22.12.X, where X indicates any number, or "don't care".	PERMIT UDP ANY HOST 30.22.21.X
Deny all packets having source IP address of 23.20.7.0 and any destination address (indicated by address X.X.X.X, where X indicates any number, or "don't care").	DENY TCP 23.20.7.0 X.X.X.X.
Permit TCP protocol packets from any source IP address going to host having an IP address of 28.16.31.10.	PERMIT TCP ANY HOST 28.16.31.10

Example of the Creation of an Exemplar Bit String Having One Field for Each Packet Header Field Utilized By at Least One ACL Rule in the ACL Rule Set, and the Subsequent Creation of Bit Strings for Each ACL Rule in the ACL Rule Set Based on the Created Exemplar

577321 v3

FIG. 7B

Example of the Creation of a Bit Selection Vector

FIG. 7C1

Count Row, with such examined Smaller Total Count row columns being those corresponding to the Larger Total Count Row columns designated as Potential, "P," candidate columns; redesignate as potential, "R," candidate columns which might be utilized as the pointers of the Bit Selection Vector, those examined Smaller Total Count row columns with the smallest number entries	K KK Since the number or redesignated potential candidates, "R," is less than the Number of Unspecified Pointers of Bit Selection Vector, Designate all redesignated, "R," candidates as Actual, "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector.	Note: The Number of Unspecified Pointers of Bit Selection Vector is currently equal to 4, and the number of redesignated potential candidates, "R," is 3, which is less than the Number of Unspecified Pointers of Bit Selection Vector; thus, all "R" potential candidates are specified Actual, "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector.
Subtract the number of Specified Actual, "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector.	Number of Unspecified Pointers of Bit Selection Vector = Number of Specified Actual, "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector Specified in Preceding Step(i.e., 3) = 1 pointer left unspecified	
Since the number of Unspecified Pointers of Bit Selection Vector is still non-zero, Mark Specified "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by	* **	Note: Row columns 1, 3, and 4 are marked with asterisks to indicate that since the these row columns have already been designated as candidates "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector.

pointers of the Bit Selection Vector with asterisks indicating that such columns are no longer selectable or under consideration, since the bit positions associated with the "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector have already been specified.

Thereafter, repeat the "select the row entries in the "Larger Total Count" row having smallest number entries . . ." operation above upon the row columns which have not yet been designated as candidate "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector

P P P P

* *

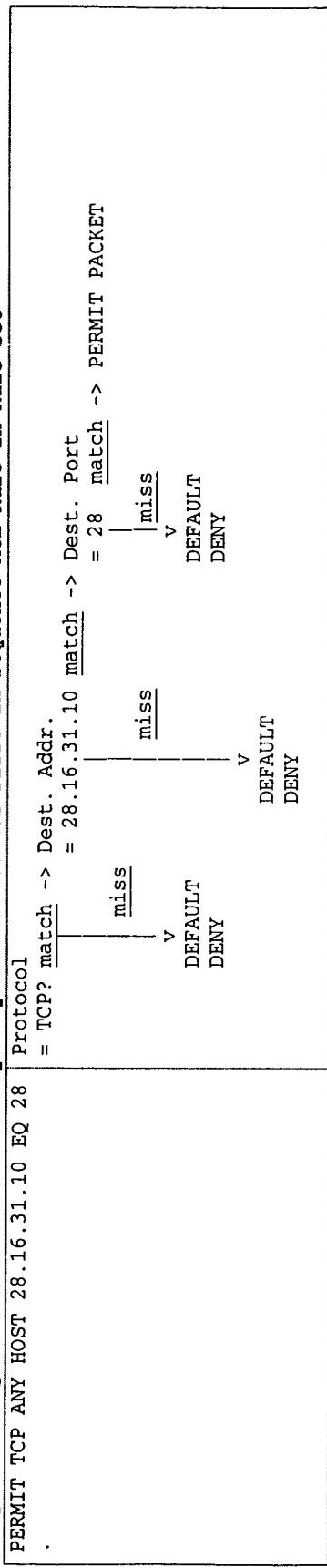
Note: Row columns 1, 3, and 4 are marked with asterisks to indicate that since the bit positions associated with these row columns have already been designated as candidates.

<p>Since there are more candidates, "P," than Number of Unspecified Pointers of Bit Selection Vector (at this point, 3 pointers have been specified as "K," meaning that one additional pointer is necessary to have the pointers required to completely point out the 4 bit hash table index), repeat the refine the selection . . ." operation above</p>	<p>Since after redesignation there are still more candidates "R" than the Number of Unspecified Pointers of Bit Selection Vector, all "R" candidates are deemed equally good choices; consequently, the number of Actual, "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector necessary to completely point out the hash table index value (i.e., in the present example, one more pointer is needed) may be selected at random from the designated "R" row columns.</p>	<p>There are now specified Actual, "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector equal in number to the bit length of the hash table index; consequently, all pointers of the Bit Selection Vector, which will be utilized to point to bit positions used to form a hash table index value which will be used to "key into"</p>	<p>R RR R R Note: Since all entries in the "Smaller Total Count" row columns, corresponding with the selected row columns of the "Larger Total Count" row, were the same number (i.e., the base ten number "3"), all P row columns are redesignated as candidates "R." K Note: Select row column 34 at random. Note: Select row column 34 at random.</p> <p>K Note: These Actual, "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector indicate that the first, third, and fourth leftmost bit positions within the "protocol ID" field, and the fourth leftmost bit positions within the "destination address" field will be utilized as the hash table index bits.</p>
--	--	---	--

the hash table, have been fully specified.	Definition of the Bit Selection Vector = Bit Selection Vector = [pointer to first leftmost bit position within the "protocol ID" field; pointer to third leftmost bit position within the "protocol ID" field; pointer to fourth leftmost bit position within the "protocol ID" field; pointer to fourth leftmost bit position within the "destination address" field]
--	---

Example Showing the Construction of Balanced Hash Table Of ACL Binary Comparison Trees

Example showing the Creation of a Binary Comparison Tree for First In Sequence ACT Rule in Rule Set



Example Showing the Addition of a Binary Comparison Tree Constructed for the First in Sequence Rule In ACL Rule Set Into The Hash Table

Select bit string constructed from first ACL rule in Rule Set, utilizing the contents of those bit positions (1, 3, 4, and 34) pointed at by the Hash-Table-Balancing Bit Selection Vector, enter hash table at entry corresponding to the bits at bit positions serving as hash key index (e.g., bit position 1 contains "0"; bit position 3 contains "0"; bit position 4 contains "0"; and bit position 34 contains "0") and build binary Comparison Tree indicative of this first selected ACL rule

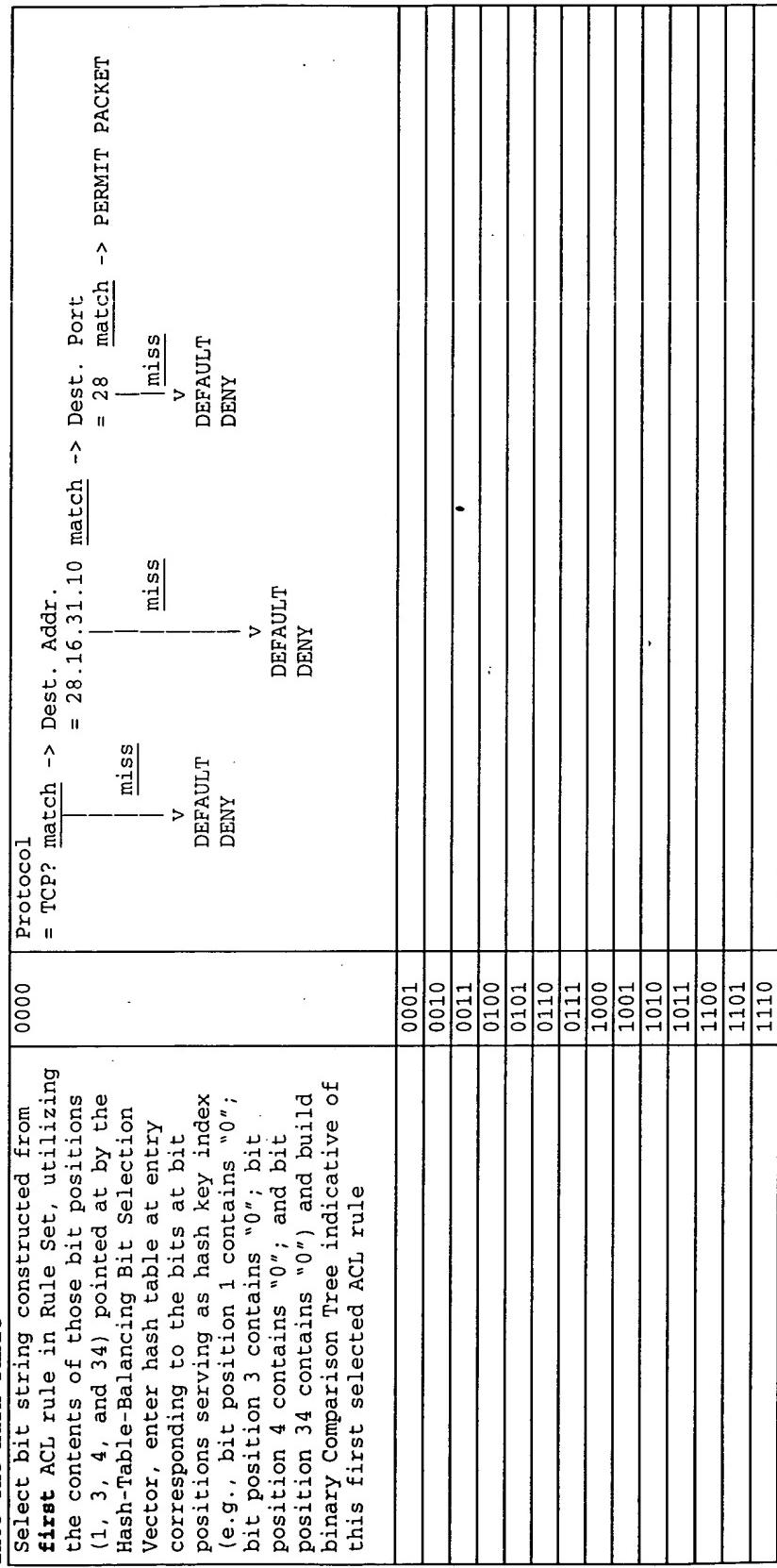


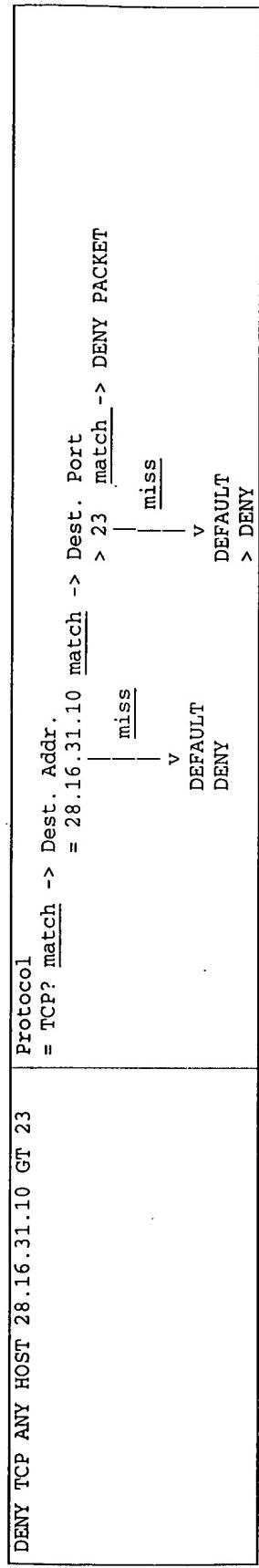


FIG. 7D2

577321 v3

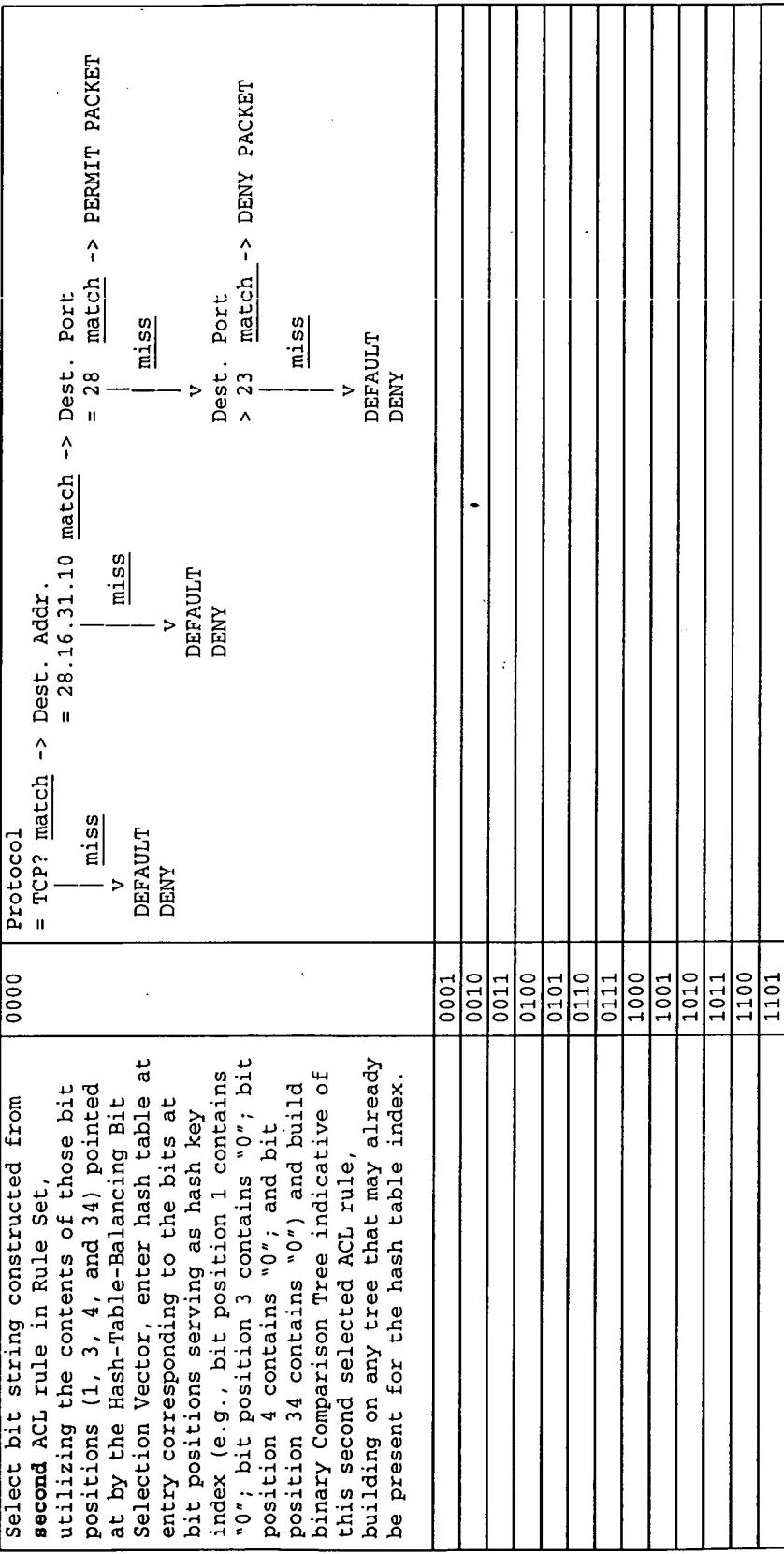
Example Showing the Construction of Balanced Hash Table of ACL Binary Comparison Trees (cont.)

Example showing the creation of a Binary Comparison Tree for Second In Sequence Rule in Rule Set



Example Showing the Addition of a Binary Comparison Tree Constructed for the Second In Sequence Rule In ACL Rule set Into The Hash Table

Select bit string constructed from **second** ACL rule in Rule Set, utilizing the contents of those bit positions (1, 3, 4, and 34) pointed at by the Hash-Table-Balancing Bit Selection Vector, enter hash table entry corresponding to the bits at bit positions serving as hash key index (e.g., bit position 1 contains "0"; bit position 3 contains "0"; bit position 4 contains "0"; and bit position 34 contains "0") and build binary Comparison Tree indicative of this second selected ACL rule, building on any tree that may already be present for the hash table index.

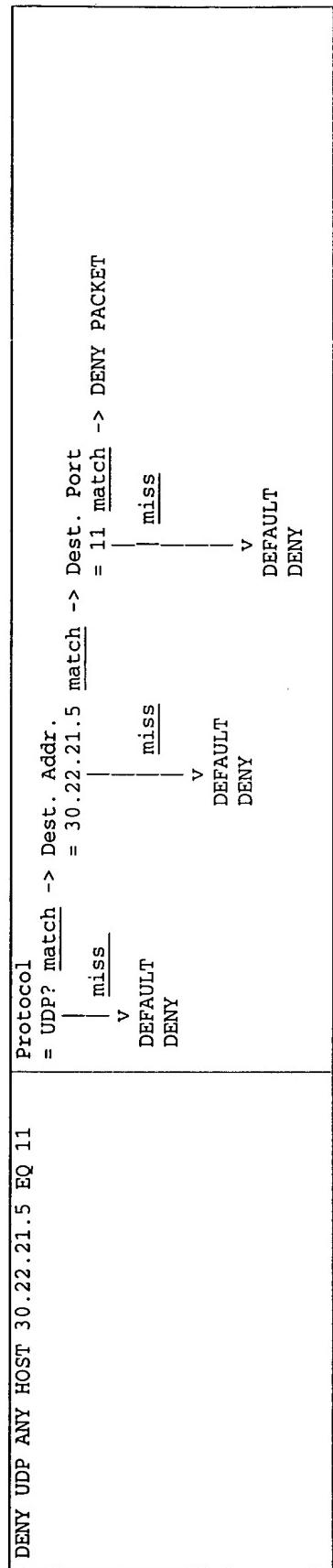


ପ୍ରକାଶକ ମେଳିକା

1110
1111

Example Showing the Construction of Balanced Hash Table of ACL Binary Comparison Trees (cont.)

Example Showing the Creation of a Binary Comparison Tree for Third In Sequence ACL Rule in Rule Set

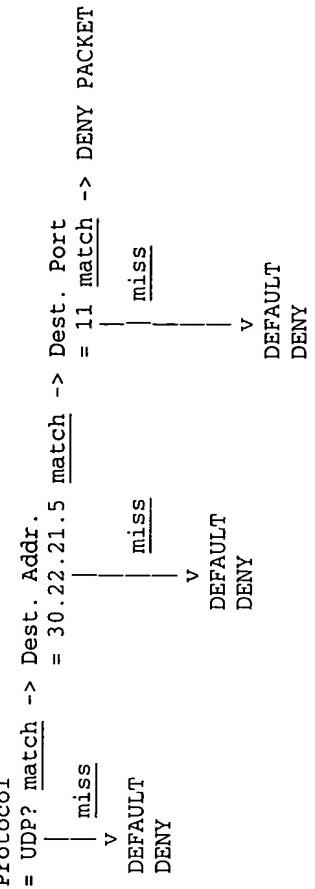


Example Showing the Addition of a Binary Comparison Tree Constructed for the Third In Sequence Rule In ACL Rule set Into The Hash Table

0001	Protocol = TCP? <u>match</u> -> Dest. Addr.
0010	= 28.16.31.10 <u>match</u> -> Dest. Port
0011	= 28 <u>match</u> -> PERMIT PACKET
0100	Dest. Port
0101	> 23 <u>match</u> -> DENY PACKET
0110	miss
0111	miss
1000	miss
1001	miss
1010	miss
1011	miss
1100	miss
1101	miss
1111	miss
1011	miss
1001	miss
1010	miss
1011	miss
1100	miss
1101	miss
1111	miss

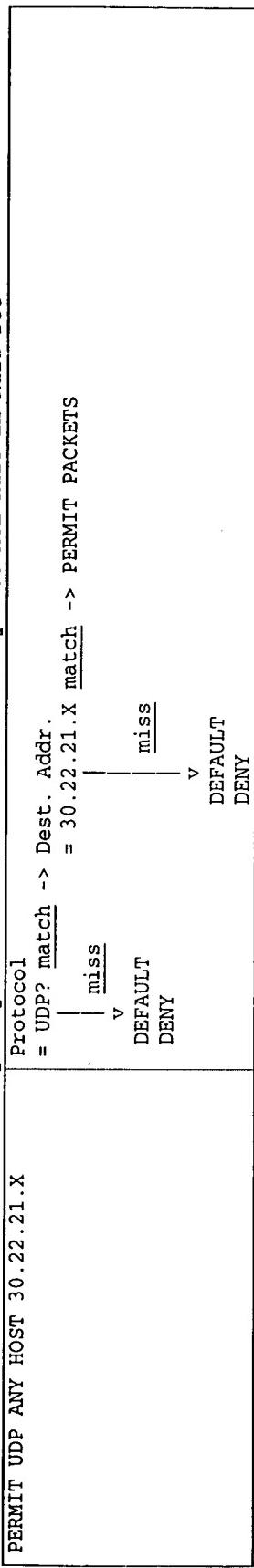
		1100	
		1101	
		1110	

Select bit string constructed from **third** ACL rule in Rule Set, utilizing the contents of those bit positions (1, 3, 4, and 34) pointed at by the Hash-Table-Balancing Bit Selection Vector, enter hash table at entry corresponding to the bits at bit positions serving as hash key index (e.g., bit position 1 contains "1"; bit position 3 contains "1"; bit position 4 contains "1"; and bit position 34 contains "1") and build binary Comparison Tree indicative of this third selected ACL rule



Example Showing the Construction of Balanced Hash Table of ACL Binary Comparison Trees (cont.)

Example Showing the Creation of a Binary Comparison Tree for Fourth In Sequence ACL Rule in Rule Set



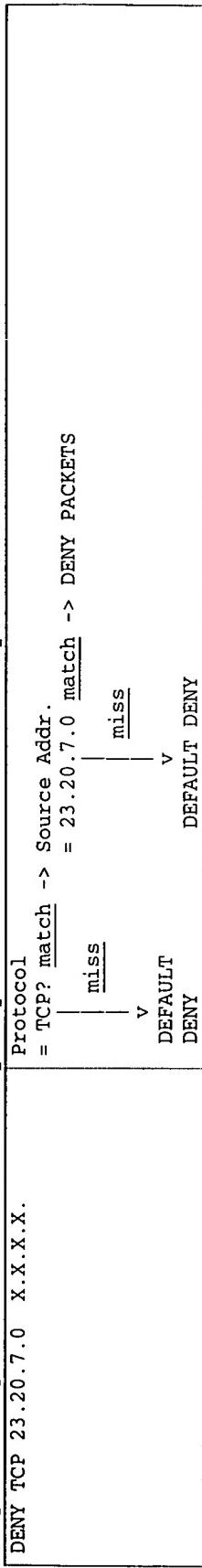
Example Showing the Addition of a Binary Comparison Tree Constructed for the Fourth In Sequence Rule In ACL Rule set Into The Hash Table

	0000	Protocol = TCP? <u>match</u> -> Dest. Addr. miss V DEFAULT DENY
		= 28.16.31.10 <u>match</u> -> Dest. Port miss V DEFAULT DENY
		= 28 <u>match</u> -> PERMIT PACKET miss V DEFAULT DENY
	0001	Dest. Port > 23 <u>match</u> -> DENY PACKET miss V DEFAULT DENY
	0010	
	0011	*
	0100	
	0101	
	0110	
	0111	
	1000	
	1001	
	1010	
	1011	
	1100	
	1101	
	1110	

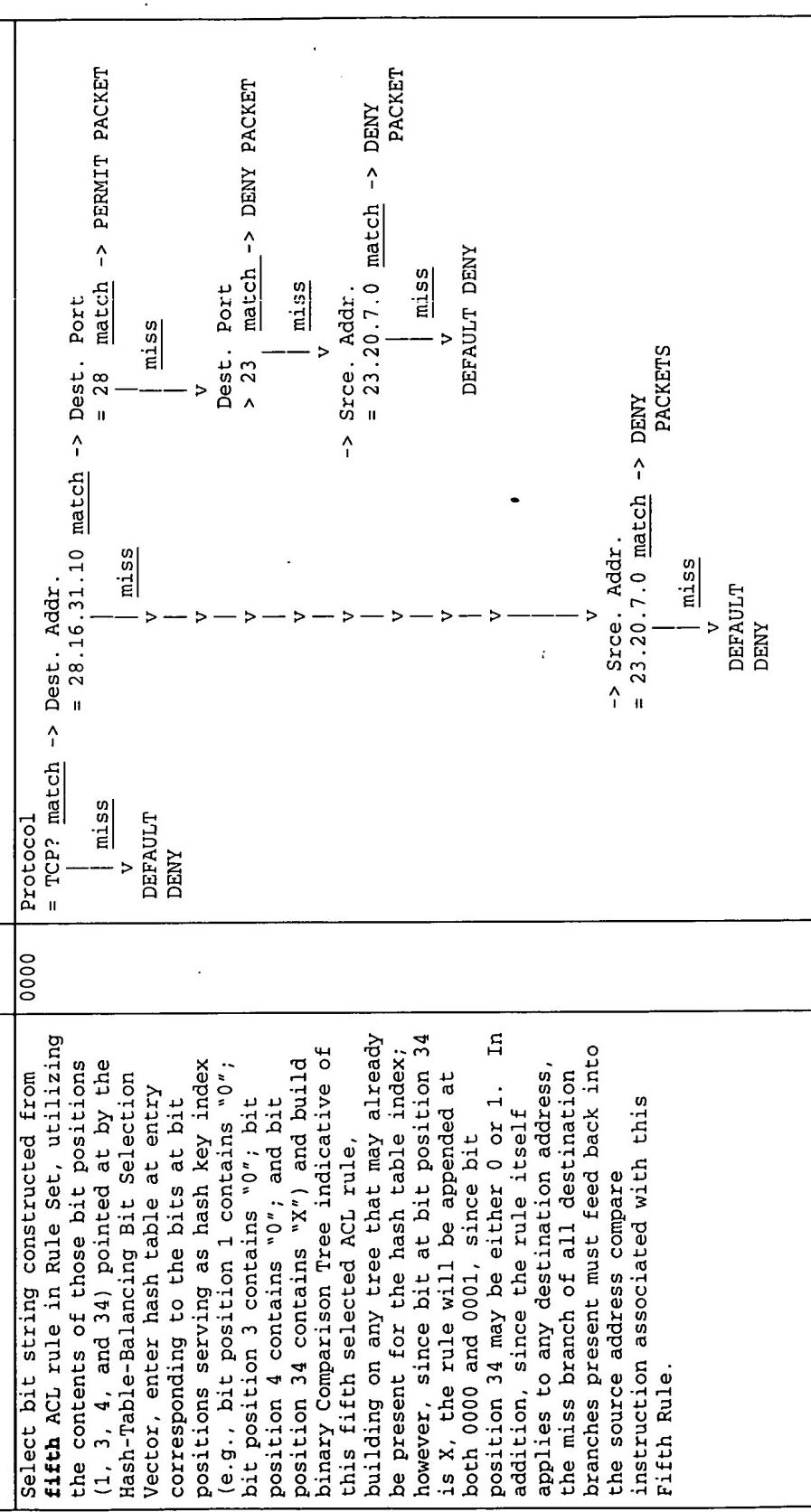
Select bit string constructed from fourth ACL rule in Rule Set, utilizing the contents of those bit positions (1, 3, 4, and 34) pointed at by the Hash-Table-Balancing Bit Selection Vector, enter hash table at entry corresponding to the bits at bit positions serving as hash key index (e.g., bit position 1 contains "1"; bit position 3 contains "1"; bit position 4 contains "1"; and bit position 34 contains "1") and build binary Comparison Tree indicative of this fourth selected ACL rule, building on any tree that may already be present for the hash table index	1111	Protocol = UDP? <u>match</u> -> Dest. Addr.
		miss v DEFAULT DENY

Example showing the Construction of Balanced Hash Table of ACL Binary Comparison Trees (cont.)

Example showing the Creation of a Binary Comparison Tree for Fifth In Sequence ACL Rule in Rule Set



Example Showing the Addition of a Binary Comparison Tree Constructed for the Fifth In Sequence Rule In ACL Rule set Into The Hash Table

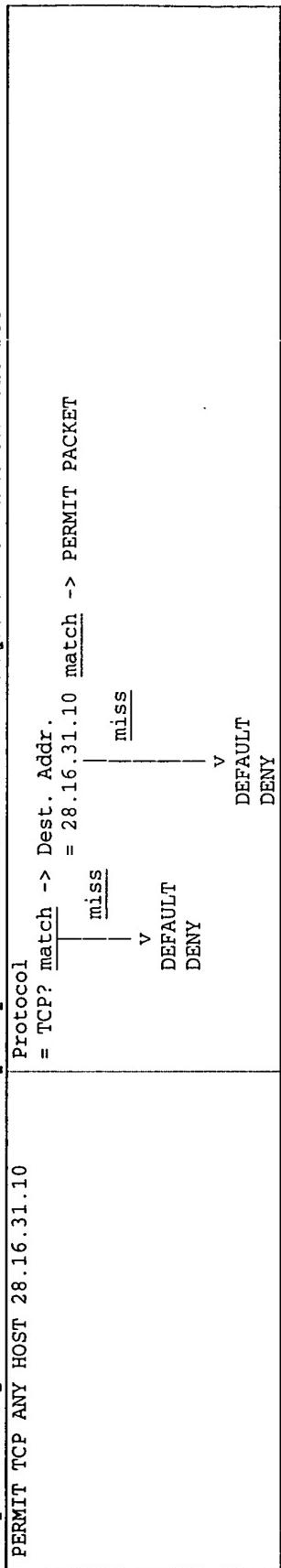


<p>Select bit string constructed from fifth ACL rule in Rule Set, utilizing the contents of those bit positions (1, 3, 4, and 34) pointed at by the Hash-Table-Balancing Bit Selection Vector, enter hash table at entry corresponding to the bits at bit positions serving as hash key index (e.g., bit position 1 contains "0"; bit position 3 contains "0"; bit position 4 contains "0"; and bit position 34 contains "X") and build binary Comparison Tree indicative of this fifth selected ACL rule, building on any tree that may already be present for the hash table index; however, since bit at bit position 34 is X, the rule will be appended at both 0000 and 0001, since bit position 34 may be either 0 or 1.</p>	<table border="1"> <tbody> <tr> <td>0001</td><td>Protocol = TCP? <u>match</u> -> Source Addr. miss V DEFAULT DENY</td><td>Protocol = 23.20.7.0 <u>match</u> -> DENY PACKETS miss V DEFAULT DENY</td></tr> <tr> <td>0010</td><td></td><td></td></tr> <tr> <td>0011</td><td></td><td></td></tr> <tr> <td>0100</td><td></td><td></td></tr> <tr> <td>0101</td><td></td><td></td></tr> <tr> <td>0110</td><td></td><td></td></tr> <tr> <td>0111</td><td></td><td></td></tr> <tr> <td>1000</td><td></td><td></td></tr> <tr> <td>1001</td><td></td><td></td></tr> <tr> <td>1010</td><td></td><td></td></tr> <tr> <td>1011</td><td></td><td></td></tr> <tr> <td>1100</td><td></td><td></td></tr> <tr> <td>1101</td><td></td><td></td></tr> <tr> <td>1110</td><td></td><td></td></tr> <tr> <td>1111</td><td>Protocol = UDP? <u>match</u> -> Dest. Addr. miss V DEFAULT DENY</td><td>Protocol = 30.22.21.5 <u>match</u> -> Dest. Port miss V = 11 <u>match</u> -> DENY PACKET</td></tr> <tr> <td></td><td></td><td>= 30.22.21.X <u>match</u> -> PERMIT PACKET miss V DEFAULT DENY</td></tr> </tbody> </table>	0001	Protocol = TCP? <u>match</u> -> Source Addr. miss V DEFAULT DENY	Protocol = 23.20.7.0 <u>match</u> -> DENY PACKETS miss V DEFAULT DENY	0010			0011			0100			0101			0110			0111			1000			1001			1010			1011			1100			1101			1110			1111	Protocol = UDP? <u>match</u> -> Dest. Addr. miss V DEFAULT DENY	Protocol = 30.22.21.5 <u>match</u> -> Dest. Port miss V = 11 <u>match</u> -> DENY PACKET			= 30.22.21.X <u>match</u> -> PERMIT PACKET miss V DEFAULT DENY
0001	Protocol = TCP? <u>match</u> -> Source Addr. miss V DEFAULT DENY	Protocol = 23.20.7.0 <u>match</u> -> DENY PACKETS miss V DEFAULT DENY																																															
0010																																																	
0011																																																	
0100																																																	
0101																																																	
0110																																																	
0111																																																	
1000																																																	
1001																																																	
1010																																																	
1011																																																	
1100																																																	
1101																																																	
1110																																																	
1111	Protocol = UDP? <u>match</u> -> Dest. Addr. miss V DEFAULT DENY	Protocol = 30.22.21.5 <u>match</u> -> Dest. Port miss V = 11 <u>match</u> -> DENY PACKET																																															
		= 30.22.21.X <u>match</u> -> PERMIT PACKET miss V DEFAULT DENY																																															

FIG. 7D10
577321 v3

Example Showing the Construction of Balanced Hash Table Of ACL Binary Comparison Trees (cont.)

Example Showing the Creation of a Binary Comparison Tree for Sixth In Sequence ACL Rule in Rule Set



Example Showing the Addition of a Binary Comparison Tree Constructed for the Sixth In Sequence Rule In ACL Rule set Into The Hash Table

